**Stanovení mědi metodou přídavku standardu s doplněním na konstantní objem**

Ze zásobního roztoku o koncentraci Cu2+ 1 g·l-1 připravíme pracovní roztok I o koncentraci 100 mg·l-1.

Navážením připravíme 100 ml pracovního roztoku II - 1% KCl.

Vzorek vína zahřejeme na cca 80 st.C, poté ochladíme na laboratorní teplotu.

Z pracovního roztoku I připravíme ředěním do 25ml odměrných baněk roztoky o následujících koncentracích mědi: 0,0 mg.l-1; 0,2 mg.l-1; 0,4 mg.l-1; 0,6 mg.l-1; 0,8 mg.l-1; 1,0 mg.l-1 a 1,2 mg.l-1. Do každé z odměrných baněk přidáme 5 ml nezředěného vzorku vína, napipetujeme takové množství 1% KCl, aby jeho obsah v baňce byl 0,1% a doplníme 1% HNO3.

Jako blank použijeme roztok 1% HNO3. Každý roztok proměříme **7krát** a hodnoty absorbance přepíšeme do Excelu na samostatný list a uložíme ve formátu **txt** pod názvem **prid\_st** do složky kalibrace. **Dbáme na to, abychom místo desetinných čárek používali tečky**. Jako blank použijeme roztok bez přídavku upravený stejným způsobem jako kalibrační roztoky.

Je vhodné data přepsat do Excelu způsobem, který zobrazuje následující tabulka:

Tab. č. 6: Data získaná z měření pomocí metody přídavku standardu

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **blank** | **0 mg/l** | **0.2 mg/l** | **0.4 mg/l** | **0.6 mg/l** | **0.8 mg/l** | **1.0 mg/l** | **1.2 mg/l** |
| 0.003411 | 0.005832 | 0.008668 | 0.01149 | 0.01452 | 0.01686 | 0.02094 | 0.02291 |
| 0.00362 | 0.005789 | 0.008652 | 0.01144 | 0.01413 | 0.01687 | 0.02039 | 0.02302 |
| 0.003653 | 0.005916 | 0.009171 | 0.01123 | 0.01401 | 0.01656 | 0.02083 | 0.02407 |
| 0.003467 | 0.005946 | 0.008877 | 0.01131 | 0.01411 | 0.01730 | 0.02052 | 0.02334 |
| 0.003153 | 0.005714 | 0.009624 | 0.01149 | 0.01501 | 0.01696 | 0.02035 | 0.02358 |
| 0.003489 | 0.005509 | 0.009425 | 0.01159 | 0.01454 | 0.01766 | 0.02086 | 0.02309 |
| 0.003744 | 0.005717 | 0.009619 | 0.01182 | 0.01472 | 0.01725 | 0.02030 | 0.02306 |

Můžeme pokračovat se zpracováním v R Studiu. Pokud není v záložkách příkazového řádku otevřen soubor prid\_st, klikneme na „File“, zvolíme „Open File“ a ze složky kalibrace vybereme soubor prid\_st.R. Před začátkém práce je opět vhodné smazat data z prostředí (Enviroment -> Clear) a z oblasti grafů (Plots -> Clear all).

Myší modře označíme první řádek skriptu a klikneme na RUN, nebo zmáčkneme Ctrl+R, tím načteme vstupní data.

* *prid\_st0<-read.delim("c:\\kalibrace\\prid\_st.txt",header=TRUE)*

Pokud jsme vstupní data ukládali bez hlavičky, přepíšeme ve skriptu header=FALSE. Opět umístění souboru (červený text) se může lišit. Kliknutím na název v okně prostředí zkontrolujeme načtená data.

Poté načteme vektor koncentrace

* *konc2<-c(seq(0,3,by=0.5)*

Dále musíme ze vstupní matice vyčlenit blank, který je v prvním sloupci,

* *blank2<-prid\_st0[,1];blank2*

a z něj vypočítat průměr

* *mean(blank2)*

dále blank z původní matice odstraníme:

* *prid\_st1<-prid\_st0[,-1];prid\_st1*

Nyní můžeme od vstupní matice odečíst blank a dostaneme tak data ponížená o tuto hodnotu:

* *ps\_blankless<-prid\_st1[1:length(prid\_st1)]-mean(blank2);ps\_blankless*

Nakonec vypočítáme z každého sloupce průměr, který použijeme pro zobrazení průměrných hodnot v grafu kalibrační přímky:

* *prumer\_blankless2<-apply(ps\_blankless,2,mean);prumer\_blankless2*

Nyní musíme data otestovat na odlehlost pomocí Grubbsova testu, ale testujeme data pro lineární závislost, takže napřed musíme tuto závislost vytvořit. V následujícím skriptu se vstupní data uspořádají do dvou sloupců a díky tomu budeme dále schopni udělat regresi pro všechna vstupní data. Označíme následující část skriptu a spustíme:

* *a<-length(ps\_blankless[1,]);a*
* *b<-length(ps\_blankless[,1]);b*
* *X<-rep(konc2,b);X*
* *Y<-ps\_blankless[1,]*
* *for(i in 2:b){*

*YY<ps\_blankless[i,]*

*Y<-as.numeric(c(Y,YY))};Y*

Označíme další řádek, kterým R Studiu zadáme příkaz, že data jsou lineárního charakteru:

* *linear<-lm(Y~X);linear*

přivoláme potřebnou knihovnu

* *library(car)*

a provedeme samotné testování odlehlosti

* *outlierTest(linear,cutoff=0.05, n.max=10, order=TRUE)*

Pokud jsou data v pořádku, zobrazí se v okně konzoly:

No Studentized residuals with Bonferroni p<0,05

Ale zobrazí se největší hodnota. Pokud zjistíme, že nějaká hodnota je odlehlá, poznamenáme si ji a uvedeme do protokolu.

Nyní provedeme testování úseku a směrnice dle nulových hypotéz *a* = 0 a *b* = 1, přivoláme potřebnou knihovnu a provedeme testování

* library(lmtest)
* coeftest(linear, vcov. = NULL, df = NULL)

Zobrazí se tabulka

t test of coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.0023536 0.0001966 11.972 7.171e-05 \*\*\*

konc2 0.1447293 0.0027263 53.086 4.485e-08 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Pravděpodobnosti pro úsek i směrnici by se měly blížit nule. Pokud ne, měření zopakujeme. Hodnoty úseku („Intercept“) a směrnice („konc“) si poznamenáme a přepíšeme v legendě, aby odpovídaly kalibrační přímce pro naše data.

Dále zobrazíme graf kalibrační přímky

* *plot(X,Y,type="n",main="Kalibrační přímka",xlab="Koncentrace Cu (mg.l^(-1))",ylab="Absorbance (a.u.)",col=3)*

Zobrazí se prázdný rámeček. Podle toho změníme umístění legendy. Fialová hodnota musí odpovídat nejmenší hodnotě na ose x a zelená hodnota zase největší hodnotě na ose y.

* *legend(0.0,0.016,legend=c("kalibrační přímka - y = 0,0024 + 0,1447x", "chybové úsečky - int. spolehlivosti"),col=c(4,"red"),box.lwd = 0,box.col = "white",bg = "white",lty=c(1,1),cex=0.9,lwd=1.5,pch=c("-","-"))*
* *abline(linear,col=4)*
* *points(konc2,prumer\_blankless2,col=1)*

do grafu ale musíme přidat chybové úsečky, pro jejich výpočet potřebujeme směrodatnou odchylku z každého sloupce

* *sm.odch\_blankless2<-apply(ps\_blankless,2,sd);sm.odch\_blankless2*

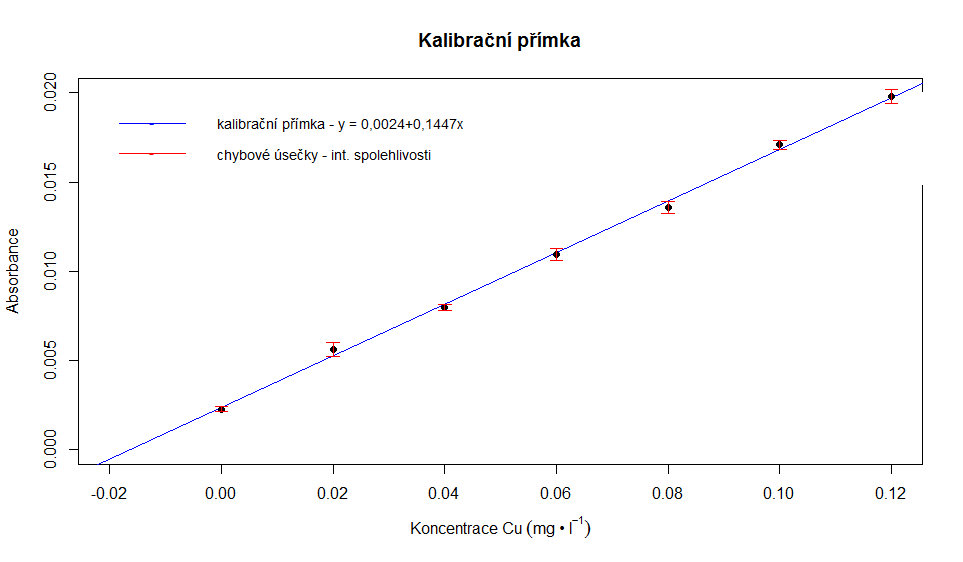
a hodnotu, kterou budeme přičítat a odečítat k průměrným hodnotám (směrodatná odchylka vynásobena kritickou hodnotou Studentova rozdělení pro *ʋ* = *n* - 1, to celé poděleno odmocninou z *n*, což je počet měření pro každý kalibrační bod):

* *L2<-((sm.odch\_blankless2\*qt(0.975,6))/sqrt(length(blank2)));L2*

Přivoláme potřebnou knihovnu a zobrazíme chybové úsečky

* *library(Hmisc)*
* *errbar(konc2, prumer\_blankless2, yplus=(prumer\_blankless2+L2), yminus=(prumer\_blankless2-L2), cap=0.015, add=T, errbar.col="red")*

Graf uložíme a přidáme do protokolu.

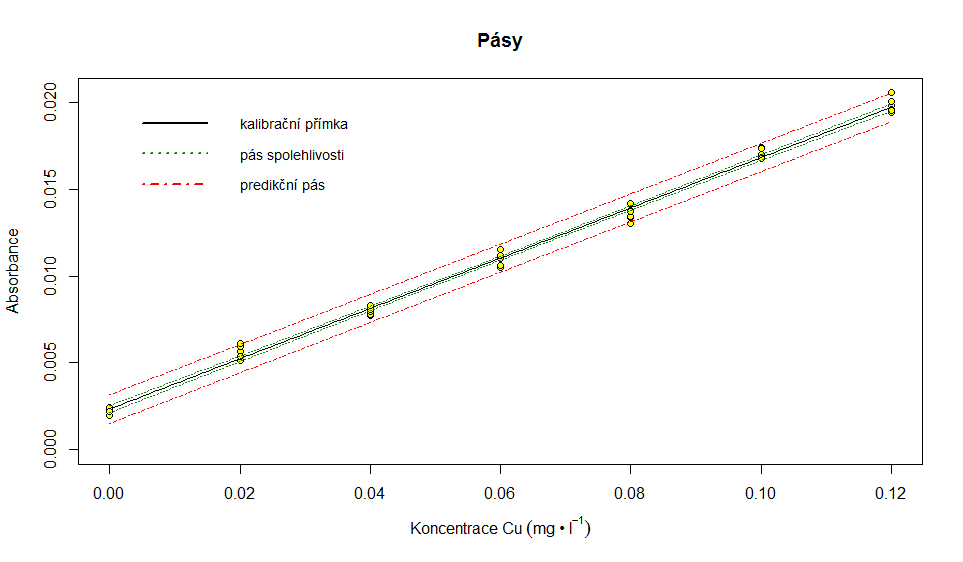


Obrázek č 14: Kalibrační přímka a chybové úsečky pro metodu přídavku

Nyní zobrazíme pásy spolehlivosti a predikce

* *library(chemCal)*
* *source("C:\\kalibrace\\calplot\_primka.r")*
* *linear2<-calplot\_primka(linear, xlim = c("auto", "auto"), ylim = c("auto", "auto"), xlab="Koncentrace Cu (mg.l^(-1))",ylab="Absorbance (a.u.)", alpha=0.05, varfunc = NULL)*

Graf uložíme a přiložíme do protokolu.



Obrázek č. 15: Pásy spolehlivosti a predikce

Poté vypočítáme mez detekce a stanovitelnosti, opět nás zajímají hodnoty na ose x, tudíž odečítáme první čísla, což odpovídá koncentraci.

* *lod(linear)*
* *> ld<-as.numeric(lod(linear));ld*

[1] 0.009460677 0.003722824

* *> lq<-(ld\*10/3);lq*

[1] 0.03153559 0.01240941

Hodnoty poznamenáme a uvedeme do protokolu.

Pokračujeme výpočtem intervalů spolehlivosti pro parametry. K tomu opět potřebujeme směrodatné odchylky úseku a směrnice

* *coeftest(linear, vcov. = NULL, df = NULL)*

které ale musíme zobrazit jako tabulku a jednotlivé hodnoty zobrazit samostatně

* *pint2<-as.matrix(coeftest(linear, vcov. = NULL, df = NULL));pint2*
* *pint2\_a1<-pint2[1,1];pint2\_a1*
* *pint2\_a2<-pint2[1,2];pint2\_a2*
* *pint2\_b1<-pint2[2,1];pint2\_b1*
* *pint2\_b2<-pint2[2,2];pint2\_b2*

Směrodatné odchylky zaokrouhlíme na dvě platné číslice a podle počtu desetinných míst zaokrouhlíme výsledné intervaly:

* *pa2<-signif(pint2\_a2, digits=2);pa2*
* *pb2<-signif(pint2\_b2, digits=2);pb2*

Dostaneme např.: pa2 = 0.00010 a pb2 = 0.0014

Pokračujeme samotným výpočtem

* *La<-c(pint2\_a1-pint2\_a2\*qt(0.975,5),pint2\_a1+pint2\_a2\*qt(0.975,5));La*
* *Lb<-c(pint2\_b1-pint2\_b2\*qt(0.975,5),pint2\_b1+pint2\_b2\*qt(0.975,5));Lb*

Podle desetinných míst u směrodatných odchylek intervaly zaokrouhlíme

* L2a<-round(La, digits=5);L1a
* L2b<-round(Lb, digits=4);L1b

Zobrazené intervaly uvedeme do protokolu jako

L(a)1,2 = <0.00210;0.00261>

L(b)1,2 = <0.1411;0.1483>

Poté zobrazíme elipsu pro parametry regresní přímky

* *library(ellipse)*
* *level = 0.975*
* *ell <- ellipse(linear, which = c(1, 2), level = level)*
* *plot(ell, type = "l", main="Elipsa pro parametry", ylab="Směrnice", xlab="Úsek")*

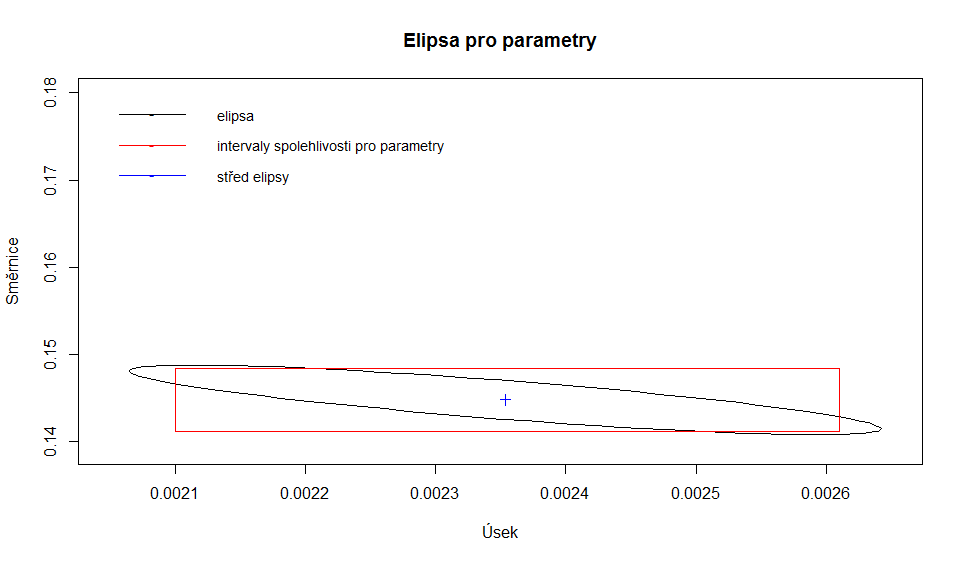
Zobrazí se elipsa, musíme ovšem upravit zobrazovanou oblast. Potřebujeme si udělat místo pro legendu nad elipsou, takže pokud se nám na ose y zobrazí oblast například od 0.10 po 0.11, v dalším řádku upravíme barevná čísla v závorce tak, aby fialové odpovídalo zobrazené nejnižší hodnotě na ose y a zelené zvolíme například o jednu setinu větší, tedy 0.12.

* *plot(ell, type = "l",main="Elipsa pro parametry", ylab="Směrnice", xlab="Úsek", ylim = c(0.10, 0.12))*

Přidáme legendu, opět upravíme čísla pozice legendy tak, aby byla v grafu vidět. Fialové číslo odpovídá nejnižší hodnotě na ose x a zelené odpovídá nejvyšší hodnotě na ose y:

* *legend(0.0039,0.09 ,legend=c("elipsa","intervaly spolehlivosti pro parametry"), col=c(1,2),box.lwd = 0,box.col = "white",bg = "white", lty=c(1,1) ,cex=0.9, lwd=1.5, pch=c("-","-"))*
* *points(linear$coefficients[1], linear$coefficients[2],pch=3)*
* *segments(L2a[1],L2b[1],L2a[2],L2b[1],col=2)*
* *segments(L2a[1],L2b[2],L2a[2],L2b[2],col=2)*
* *segments(L2a[1],L2b[1],L2a[1],L2b[2],col=2)*
* *segments(L2a[2],L2b[2],L2a[2],L2b[1],col=2)*

Graf uložíme a přidáme do protokolu



Obrázek č. 16: Elipsa pro parametry regresní přímky

Nyní zobrazíme elipsy spolehlivosti a predikce

* *level <- 0.95*
* *shape <- var(cbind(X, Y))*
* *center <- c(mean(konc2), mean(prumer\_blankless2))*
* *radiusC <- sqrt(2 \*(length(konc2)-1)\* qf(level, 2, (length(konc2) - 2))/(length(konc2)\*(length(konc2)-2)))*
* *radiusP <- radiusC\*sqrt(length(konc2)+1)*
* *ellcalc <- function(center, shape, radius, segments=1000){*

*segments=segments*

*angles <<- seq(from=0,to=(2\*3.1415926),by=1/segments)*

*unit.circle <<- cbind(cos(angles), sin(angles))*

*ELLIPSE <<- t(center + radius \* t(unit.circle %\*% chol(shape)))*

*return(ELLIPSE)}*

* *ellC<- ellcalc(center, shape, radiusC)*
* *ellP<- ellcalc(center, shape, radiusP)*
* *plot(X, Y, type="p",main="elipsy",xlab="Koncentrace Cu (mg.l^(-1))", ylab="Absorbance (a.u.)")*

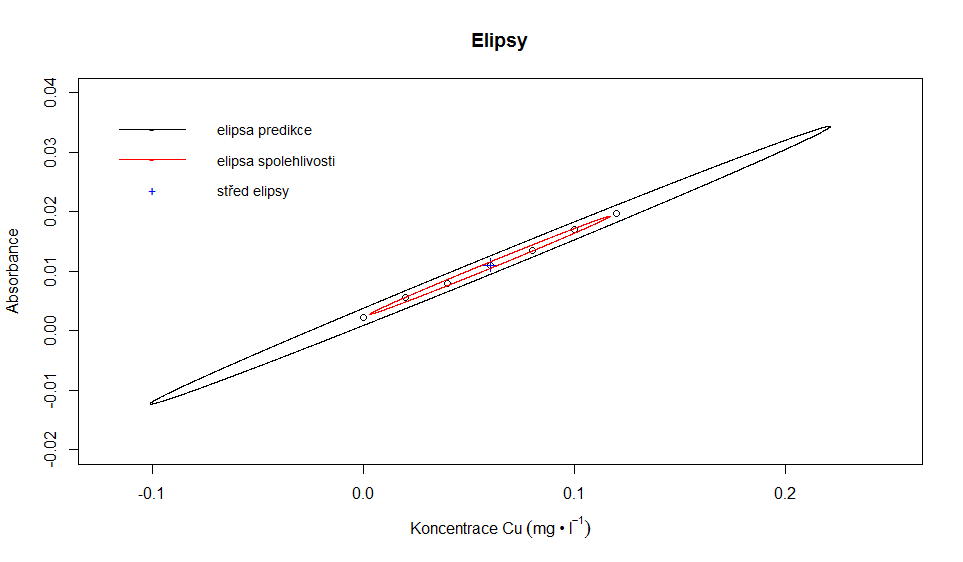
Zobrazí se pouze rámeček grafu. Nyní musíme upravit zobrazovanou oblast tak, aby elipsy byly vidět celé. Toho dosáhneme tak, že změníme hodnoty v rámečku tak, aby odpovídali trojnásobku zobrazované vzdálenosti rámečku. Pokud se nám na ose x zobrazí 0 - 0.12, nastavíme hodnoty *xlim* na -0,12 a 0,24. obdobně postupujeme u osy y a hodnot *ylim*.

* *plot(X,Y, type="n",main="elipsy",* *xlab=expression(Koncentrace ~ Cu ~ (mg ~ "\*" ~l^{-1})), ylab="Absorbance (a.u.)", xlim=c(-0.12,0.25),ylim=c(-0.02,0.04))*

Dále zobrazíme legendu. Pozici legendy musíme upravit podle zobrazované oblasti, z *xlim* vybereme první souřadnici (-0,12) a z *ylim* druhou (0.04), podle barev.

* *legend(-0.12,0.04,c("elipsa predikce","elipsa spolehlivosti","průměr"), col=c(1,2,4), lty=c(1,1,0),box.lwd = 0,box.col = "white",bg = "white", cex=0.9, lwd=1.5,pch=c("-","-","+"))*
* *points(konc2,prumer\_blankless2)*
* *points(center[1], center[2], pch = 3,col=4,cex=1.25)*
* *points(ellC[,1],ellC[,2],col=2,type="l")*
* *points(ellP[,1],ellP[,2],col=1,type="l")*

Obrázek uložíme a přidáme k protokolu.



Obrázek č. 16: Elipsa spolehlivosti a elipsa predikce

Nyní můžeme přikročit k výpočtu koncentrace mědi ve vzorku

* *konc.Cu<-((pint2\_a1/pint2\_b1)\*5);konc.Cu*

Pro výpočet intervalu spolehlivosti potřebujeme směrodatnou odchylku, pro její výpočet zase potřebujeme směrodatnou odchylku sy,x:

* *xpr2<-mean(konc2);xpr2*
* *ypr2<-mean(prumer\_blankless2);ypr2*
* *Q2<-sum(7\*(konc2-xpr2)^2);Q2*
* *Y2<-(pint2\_a1+konc2\*pint2\_b1);Y2*
* *n<-(length(konc2)\*length(blank2));n*
* *ps1<-ps\_blankless[1,];ps1*
* *ps2<-ps\_blankless[2,];ps2*
* *ps3<-ps\_blankless[3,];ps3*
* *ps4<-ps\_blankless[4,];ps4*
* *ps5<-ps\_blankless[5,];ps5*
* *ps6<-ps\_blankless[6,];ps6*
* *ps7<-ps\_blankless[7,];ps7*
* *psY1<-(sum((ps1-Y2)^2));psY1*
* *psY2<-(sum((ps2-Y2)^2));psY2*
* *psY3<-(sum((ps3-Y2)^2));psY3*
* *psY4<-(sum((ps4-Y2)^2));psY4*
* *psY5<-(sum((ps5-Y2)^2));psY5*
* *psY6<-(sum((ps6-Y2)^2));psY6*
* *psY7<-(sum((ps7-Y2)^2));psY7*
* *psY<-(psY1+psY2+psY3+psY4+psY5+psY6+psY7);psY*
* *syx2<-sqrt((psY)/(n-2));syx2*

Nyní můžeme vypočítat směrodatnou odchylku *sx*

* *sx<-((syx2/pint2\_b1)\*sqrt((1/n)+((0-ypr2)^2/(pint2\_b1^2\*Q2))));sx*

Směrodatnou odchylku zaokrouhlíme na dvě platné číslice a podle počtu míst za desetinnou tečkou pak také zaokrouhlíme interval spolehlivosti

* *sx1<-signif(sx, digits=2);sx1*

Vypočítáme samotný interval spolehlivosti

* *Lvz<-c(konc.Cu-qt(0.975,47)\*sx1,konc.Cu+qt(0.975,47)\*sx1);Lvz*

Interval zaokrouhlíme

* *Lvz<-round(Lv, digits=5);Lvz*

a stanovený obsah Cu2+ uvedeme do protokolu jako

Lvz = <0.07962;0.08300> mg·l-1.