

Aufgaben Woche 2

Datensatz “umsatzdaten_gekuerzt.csv” aus dem Github-Repository importieren.

Mit Hilfe eines Balkendiagramms über alle Warengruppen hinweg den Zusammenhang der durchschnittlichen Umsätze je Wochentag darstellen.

Fügt in einem zweiten Schritt zusätzlich Konfidenzintervalle der Umsätze je Wochentag hinzu. Lese Dir dazu das in der R Graph Gallery („barplot with error bars“) dargestellte Vorgehen durch und passe es auf den Datensatz mit den Umsatzdaten an.

Als zusätzliche (optionale) Aufgabe könnt Ihr versuchen, die Umsätze je Wochentag getrennt nach Warengruppe darzustellen (ein eigenes Balkendiagramm je Warengruppe), um einen genaueren Einblick in die Daten zu erhalten.

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

##
## -- Column specification -----
## cols(
##   Datum = col_date(format = ""),
##   Warengruppe = col_double(),
##   Umsatz = col_double()
## )
```

Auffüllen fehlender Umsätze

```
tage <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
warengruppe <- 1:6
datum <- c(unique(umsatzdaten$Datum))

control_df <- data.frame(Datum = as.Date(character()),
                        Warengruppe = integer(), Umsatz = integer() )

for(i in 1:length(datum)) {
  for(j in 1:6)
    if(length(umsatzdaten[umsatzdaten$Datum == as.Date(datum[i]) &
                        umsatzdaten$Warengruppe == j,]$Umsatz) == 0) {
      test_df <- data.frame(Datum = as.Date(datum[i]), Warengruppe = j, Umsatz = 0 )
      control_df <- rbind(control_df, test_df)
      umsatzdaten <- rbind(umsatzdaten, test_df)
    }
}

aktualisierte_wg <- sort(unique(control_df$Warengruppe))
cat("Umsätze aufgefüllt für Warengruppen: ", aktualisierte_wg, "\n")

## Umsätze aufgefüllt für Warengruppen:  4 6

for(i in 1:length(aktualisierte_wg)) {
  cat("Warengruppe: ", aktualisierte_wg[i], " -> ", as.character(
    count(control_df[control_df$Warengruppe == aktualisierte_wg[i],])),
    " Datensätze mit 0 Umsatz hinzugefügt\n")
}

## Warengruppe:  4  ->  54  Datensätze mit 0 Umsatz hinzugefügt
## Warengruppe:  6  -> 1771 Datensätze mit 0 Umsatz hinzugefügt

rm(i, j, aktualisierte_wg, test_df, control_df)

# Erstellung der Variable mit dem Wochentag
umsatzdaten$wochentag <- weekdays(umsatzdaten$Datum)
#Sortierreihenfolge für Wochentage festlegen
umsatzdaten$wochentag <- factor(umsatzdaten$wochentag, levels = c(
  "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))

#Gruppierung der Umsatzdaten um Gesamt pro Tag erstellen zu können
umsatzdaten <- group_by(umsatzdaten, Datum)
#Summe alle Warengruppen pro Tag
tagesumsatz <- umsatzdaten %>% summarise(summe = (sum(Umsatz)))

## `summarise()` ungrouping output (override with `.groups` argument)

#Wochentag zu Datum
tagesumsatz$wochentag <- weekdays(tagesumsatz$Datum)
#Tagesumsatz Gruppieren
tagesumsatz <- group_by(tagesumsatz, wochentag)

#Mean pro Wochentag
wochentag_daten_mean <- tagesumsatz %>% summarise(mean_per_day = mean(summe))
```

```

## `summarise()` ungrouping output (override with `.groups` argument)
wochentag_daten_sd <- tagesumsatz %>% summarise(standard_deviation = sd(summe))

## `summarise()` ungrouping output (override with `.groups` argument)
wochentag_daten_anzahl <- tagesumsatz %>% summarise(anzahl = n())

## `summarise()` ungrouping output (override with `.groups` argument)
wochentag_daten <- data.frame(wochentag = wochentag_daten_mean$wochentag,
                             tages_mean = wochentag_daten_mean$mean_per_day, tages_sd =
                             wochentag_daten_sd$standard_deviation,
                             anzahl = wochentag_daten_anzahl$anzahl)

wochentag_daten$wochentag <- factor(wochentag_daten$wochentag, levels = c(
  "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))

wochentag_daten <- wochentag_daten %>% mutate(tages_se = tages_sd / (sqrt(anzahl)))
wochentag_daten <- wochentag_daten %>% mutate(tages_ci = 1.96 * tages_se)

```

Berechnete Daten - SD SE CI

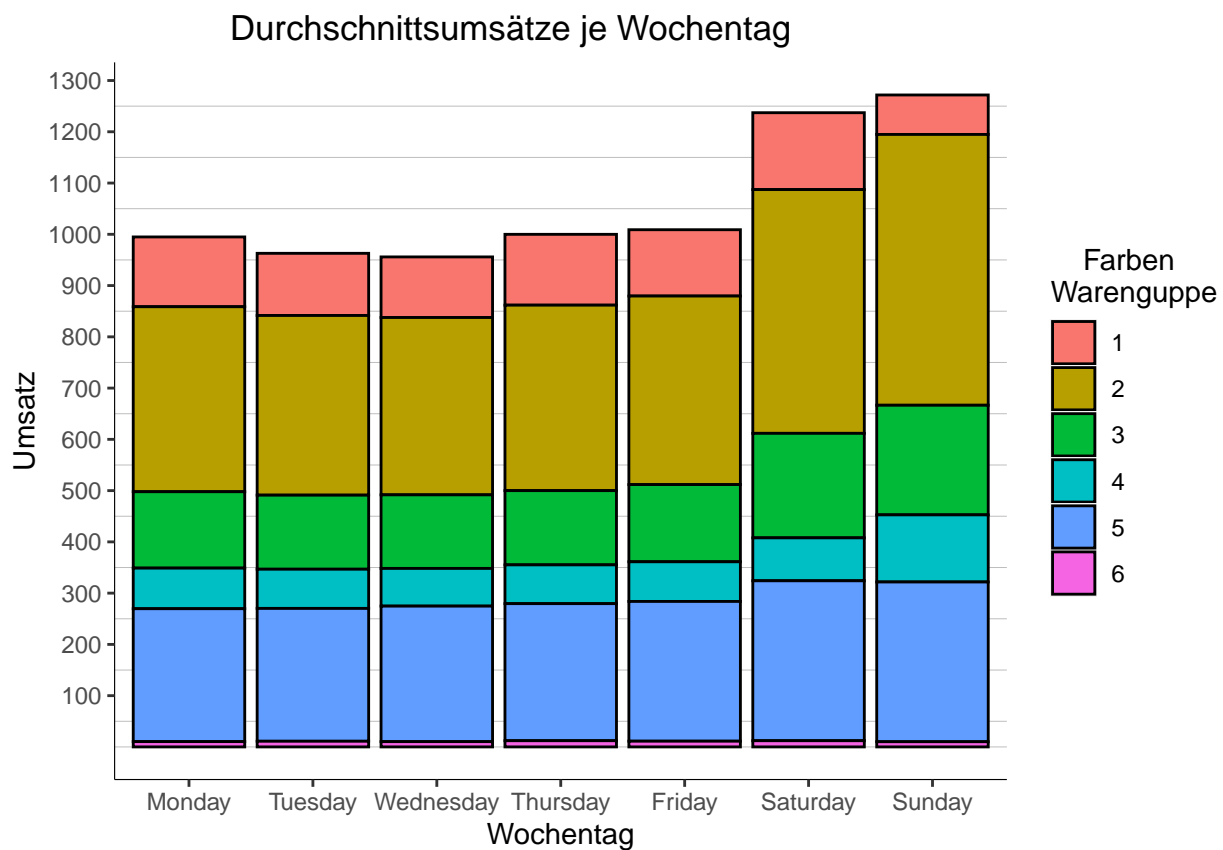
wochentag_daten

##	wochentag	tages_mean	tages_sd	anzahl	tages_se	tages_ci
## 1	Friday	1008.9374	234.5579	299	13.56483	26.58707
## 2	Monday	994.8569	290.7532	302	16.73097	32.79270
## 3	Saturday	1237.2061	295.7804	305	16.93633	33.19522
## 4	Sunday	1271.8207	221.2633	304	12.69032	24.87303
## 5	Thursday	999.9694	258.8539	303	14.87077	29.14670
## 6	Tuesday	962.9251	232.1966	304	13.31739	26.10208
## 7	Wednesday	955.8969	256.2918	302	14.74794	28.90595

```

ggplot(umsatzdaten, aes(x = wochentag, y = Umsatz, color = factor(
  Warengruppe), fill = factor(Warengruppe) )) +
  geom_bar( stat = "summary", fun = "mean", col = "black" ) +
  theme(panel.grid = element_blank(),
        panel.background = element_rect(fill = "white"),
        axis.line.x = element_line( colour = "black", size = 0.2),
        axis.line.y = element_line( colour = "black", size = 0.2),
        panel.grid.minor = element_line( colour = "grey", size = 0.1)
  ) +
  labs(fill = "    Farben \nWarenguppe") +
  ggtitle("    Durchschnittsumsätze je Wochentag") +
  scale_y_continuous("Umsatz", breaks = c(
    100,200,300,400,500,600,700,800,900,1000,1100, 1200,1300, 1400)) +
  scale_x_discrete("Wochentag")

```



```

ggplot(wochentag_daten, aes(x = wochentag, y = tages_mean)) +
  geom_bar(stat = "identity") +
  geom_text(label = as.integer(wochentag_daten$tages_mean), vjust = 10,
            colour = "#33CCCC", size = 4) +
  geom_errorbar(aes ( x = wochentag, ymin = tages_mean - tages_ci,
                     ymax = tages_mean + tages_ci), width = 0.4,
               colour = "orange", size = 0.6) +
  geom_text( aes(label = as.integer(tages_mean + tages_ci) ), vjust = -1) +
  geom_text( aes(label = as.integer(tages_mean - tages_ci) ), vjust = 3,
            colour = "white") +
  geom_text( aes(label = as.integer(tages_ci) ), vjust = -0, hjust = -2,
            colour = "red", size = 3) +
  ggtitle("          Umsätze mit Confidence Intervall") +
  xlab("Wochentag") + ylab("Umsatz") +
  theme(panel.grid = element_blank(),
        panel.background = element_rect(fill = "white"),
        axis.line.x = element_line( colour = "black", size = 0.2),
        axis.line.y = element_line( colour = "black", size = 0.2),
        panel.grid.minor = element_line( colour = "grey", size = 0.1)
  )

```

